ATID Application Development Framework Reference Manual –Modem

Revision: Ver. 0.3

Date: September, 2013

ATID Co., Ltd

Table of Contents

Acronym

modules	Descriptions
AADF	ATID Application Development Framework
SIM	Subscriber Identity Module
PIN	Personal Identification Number
PUK	Pin Unlock code
GSM	Global System for Mobile Communications
UTRAN	Universal Terrestrial Radio Access Network
PLMN	Public Land Mobile Network
IMSI	International Mobile Subscriber Identity
HLR	Home Location Register

Revision History

Version	Date	Reason	Description	Author
0.1	2012/01/17	Draft		Y. J. CHO
0.2	2013/01/27	Update	1.SimQueryCardHolderStatsu added. 2. GetRSSI added. 3. PowerEnable added.	Y. J. CHO
0.3	2013/09/13	Update	1. IsOpen added.	Y. J. CHO

1 .NET API Reference

1.1 Enumerations

1.1.1 MODEM_RESULT

The result of a call to functions.

- **MODEM_RESULT_SUCCESS**
  Function executed successfully.
- **MODEM_RESULT_OUTOFMEMORY**
  Failed to assign memory.
- **MODEM_RESULT_INVALID_ARGS**
  Invalid parameter.
- **MODEM_RESULT_UNSUPPORTED**
  Not support command currently.
- **MODEM_RESULT_FAILURE**
  Command execution failed
- **MODEM_RESULT_ALREADY_OPENED**
  Modem port already opened.
- **MODEM_RESULT_NO_BATTERY**
  Main battery insufficiency.
- **MODEM_RESULT_TIMEOUT**
  Module not response, after opened.
- **MODEM_RESULT_ALREADY_ALLOCATED**
  Modem memory already assigned.
- **MODEM_RESULT_NOT_OPENED**
  .
- **MODEM_RESULT_NOT_POWER_ON**
  Execute function without turning on the power.
- **MODEM_RESULT_ALREADY_POWER_ON**
  Modem power already turned on.

1.1.2 MODEM_SYS_INDEX_USER_WND

Flag that will be transferred by parameter while running Callback delegate.

- **MODEM_SYS_USER_WND_BASIC**

the basic function such as assign memory, power control and so on.

- **MODEM_SYS_USER_WND_CALL**

  Voice communication function.

- **MODEM_SYS_USER_WND_SMS**

  Text message function.

- **MODEM_SYS_USER_WND_SOCKET**

  Single Socket function.

- **MODEM_SYS_USER_WND_ALL**

  Whole function.

1.1.3 MODEM_SYS_NOTI

Indicate the running reason of Callback delegate.

- **NETWORK_NOT_REGISTERED**

  Cannot confirm whether access to the network for about two minutes.

- **NETWORK_REGISTERED**

  Connected with network.

- **NOCARRIER**

  Disconnected from the network.

- **PORT_CLOSE_FAIL**

  Failed to close the modem port.

- **PORT_CLOSE_SUCCESS**

  successfully closed the modem port.

- **PORT_OPEN_FAIL**

  Failed to open the modem port.

- **PORT_OPEN_SUCCESS**

  successfully opened the modem port.

- **POWER_DOWN_FAIL**

  Failed to turn off the modem power.

- **POWER_DOWN_SUCCESS**

  succeeded to turn off the modem power.

- **POWER_UP_FAIL**

  Failed to turn on the modem power.

- **POWER_UP_SUCCESS**

  succeeded to turn on the modem power.

- **POWER_UP_INITIALIZE**

  to start to turning on the modem power.

1.1.4 MODEM_POWER_STATUS

Using GetPowerStatus function to checking the status of the modem power.

- **MODEM_POWER_STATUS_OFF**
  Modem power is in the status of turned OFF.
- **MODEM_POWER_STATUS_ON**
  Modem power is in the status of turned ON.
- **MODEM_POWER_STATUS_INITIALIZING**
  Modem power is in the status of initializing (Not used)
- **MODEM_POWER_STATUS_DEINITIALIZING**
  Modem power is in the status of de-initializing (Not used)
- **MODEM_POWER_STATUS_REINITIALIZING**
  Modem power is in the status of re-initializing (Not used)

1.1.5 MODEM_SYS_PIN_AUTH_STATUS

Responding to Sim Card operation.

- **CARD_HOLDER_TRAY_REMOVED**
  SIM Card is not inserted into slot.
- **INCORRECT_PASSWORD**
  Password is incorrect.
- **INVALID_INPUT_VALUE**
  Inputted value is invalid.
- **SIM_BLOCKED**
  Aborted because of a previous operation.
- **SIM_BUSY**
  Previous operation is progressing.
- **SIM_FAILURE**
  Failed to execute SIM relevant function.
- **SIM_INSERTED**
  SIM Card inserted into slot.
- **SIM_PIN**
  the status of waiting to input PIN.
- **SIM_PUK**
  After failed to input PIN in three times, then the status of waiting to input PUK.
- **SIM_READY**
  PIN is already inputted.
- **SIM_SUCCESS**
  Succeeded to perform SIM relevant function.

- **SIM_WRONG**
  Invalid SIM Card.

1.1.6 MODEM_NETWORK_REGISTRATION_STATUS

Using GetNetworkRegistrationStatus function to checking the network connection status.

- **COMMAND_FAILED**
  failed to perform command.
- **NOT_REGISTERED**
  in the status of disconnect to network, and do not try to connect to other new network. That reason might as below:
  - no SIM card available
  - no PIN entered
  - no valid Home PLMN entry found on the SIM
- **NOT_REGISTERED_BUT_SEARCHING_OPERATOR**
  searching for aavailable network, although didn't access to network.
- **REGISTERED_TO_GSM**
  connected to GSM(2G) network.
- **REGISTERED_TO_UTRAN**
  connected to UTRAN(3G) network.
- **REGISTRATION_DENIED**
  fail to registration or authentication, The reason might as below:
  - IMSI unknown at HLR
  - illegal Mobile Station
  - illegal Mobile Equipment
- **UNKNOWN**
  unknown.

1.2 Structures

1.2.1 MODEM_CALLBACK_DATA

Using SetCallbackfunction to register the Callback delegate, then checking the structure of indicate the status changing of modem or the results of function execution.

```
Public structMODEM_CALLBACK_DATA
{
  IntPtrlParam;
  intm_UserWnMsg;
  MODEM_SYS_INDEX_USER_WNDnWndIndex;
  IntPtrwParam;
};
```

- **lParam**
  For more information on message. (Not used)
- **m_UserWnMsg**
  Indicate the call reason of callback.
- **nWndIndex**
  Indicate the call reason of callback. (Not used)
- **wParam**
  For more information on message. (Not used)

1.3 Delegates

1.3.1 ModemCallbackProc

Callback delegate which will be executed when received the structure of indicate the status changing of modem or the results of function execution.

Using ModemSetCallback(ModemCallbackProcCallbackProc) function to register ModemCallbakcProc, then Callback delegate will be executed which registered when received the structure of indicate the status changing of modem or the results of function execution.

Public delegate void ModemCallbackProc(MODEM_CALLBACK_DATACallbackData);

1.4 Methods

1.4.1 AllocContext

Modem devices are required to allocate resources.

MODEM_RESULTAllocContext();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

It must firstly to be run in order to use Modem API.

1.4.2 DeallocContext

Deallocate the modem device resources.

MODEM_RESULTDeallocContext();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if successfully deallocate resources.

Notes

When finishing using modem, deallocate resources by all means calling this function.

1.4.3 PowerUp

Turn the modem power on.

MODEM_RESULTPowerUp();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

Notes

After power is applied, it takes a few seconds to initialize Modem. When initialization

is complete, status is informed to callback delegate.

1.4.4 PowerDown

Remove the modem power.

```
MODEM_RESULTPowerDown();
```

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

Notes

When removal is complete, status will beinformed to callback delegate.

1.4.5 PowerEnable

Enabling or removingpower of Modem synchronously.

```
MODEM_RESULT    PowerEnable(
        bool Enable
);
```

Parameters

Enable

Status to be applied to modem.

True : On, False: Off

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

Notes

Unlike function of Powerup, Powerdown, function will be returned only after status of Modem power completely changed.

1.4.6 SetCallback

Setting Callback delegate which will be informed the status of modem or the results of the function execution.

```
MODEM_RESULTSetCallback(
        ModemCallbackProcCallbackProc
);
```

Parameters

CallbackProc

Callback delegate

Return Values

MODEM_RESULT_SUCCESS will be returned if function performed successfully.

Notes

In application program, in order to get the status of modem or the results of the functionExecution, you must register callback delegate using this function.

1.4.7 GetPowerStatus

Read the status of the modem power.

MODEM_POWER_STATUSGetPowerStatus();

Parameters

None

Return Values

Returning the status values of modem power(reference to MODEM_POWER_STATUS)

1.4.8 Open

Open the port which will communicate with modem, and then checking the status of the Network.

MODEM_RESULT Open();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

1.4.9 Close

Close the port which will communicate with the modem.

MODEM_RESULT Close();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

1.4.10 IsOpened

Return whether or not the modem communication port is open.

```
BOOLIsOpened()
```

Parameters

None

Return Values

True: Port is open.

False: Port is close.

1.4.11 IsOpen

Return whether or not the modem communication port is open.

Same function as IsOpened function.

```
BOOL   IsOpened()
```

Parameters

None

Return Values

True: Port is open.

False: Port is close.

1.4.12 GetRSSI

Reading the value of Modem signal intensity.

```
MODEM_RESULT   GetRSSI(
      ref int RSSI
)
```

Parameters

RSSI

Variable in which the value of Modem signal intensity will be stored.

0 : -113 dBm or less

1 : -111 dBm

2..30 : -109... -53dBm

31 : -51dBm or greater

99 : not known or not detectable

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

1.4.13 IsRegistered

Return whether or not the modem connected to network.

```
BOOLIsRegistered()
```

Parameters

None

Return Values

True: modem connected to Network.

False: modem not connected to Network.

Notes

In order to call Rasdial, there is need to check the status of network connection using this function.

1.4.14 SimChangePin

Change the SIM Card's PIN.

```
MODEM_SIM_PIN_AUTH_STATUSSimChangePin (
    stringOldPIN,
    stringNewPIN
);
```

Parameters

OldPIN

Current PIN number.

NewPIN

New PIN number

Return Values

SIM_SUCCESS will returned if performed successfully,

1.4.15 SimQueryAuthStatus

Checking the status of Network authentication.

```
MODEM_SIM_PIN_AUTH_STATUSSimQueryAuthStatus();
```

Parameters

None

Return Values

Return the status value of the Authentication.

Notes

In order to communicate, there is need to checking whether or not the SIM Card's authentication status is READY. If in the status of SIM_PIN, should enter PIN number, and if in the status of SIM_PUK, then should enter PUK number.

1.4.16 SimQueryCardHolderStatus

Returning the equipped status of SIM Card.

```
MODEM_SIM_PIN_AUTH_STATUS SimQueryCardHolderStatus();
```

Parameters

None

Return Values

Returning the equipped status of SIM Card.

Reference to 1.1.5 MODEM_SIM_PIN_AUTH_STATUS

1.4.17 SimEnterPin

Enter the SIM Card PIN.

```
MODEM_SIM_PIN_AUTH_STATUSSimEnter(
    string PIN
);
```

Parameters

PIN

SIM Card PIN number.

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

In order to communicate, there is need to using SimQueryAuthStatus function to check whether or not the status of SIM card's authentication is READY. If the execution result of SimQueryAuthStatus function is SIM_PIN, then should enter PIN number.

1.4.18 SimEnterPuk

Enter the SIM Card's PUK.

```
MODEM_SIM_PIN_AUTH_STATUSSimEnterPuk (
    stringPUK,
    stringNewPIN
);
```

Parameters

PUK

SIM Card PUK number.

NewPIN

new PIN number

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

In order to communicate, there is need to using SimQueryAuthStatusfunction to check whether or not the status of SIM card's authentication is READY. If continually entered incorrect PIN number three times, then not only should be SIM_PUK status, but also need to enter New PIN number and PUK number.

1.4.19 SimGetPinCounter

Read PIN counter

```
MODEM_SIM_PIN_AUTH_STATUSSimGetPinCounter (
    refintPinCounter
);
```

Parameters

PinCounter

the remaining frequency of enter PIN number.

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

PinCounter will reduce one time, every time enter incorrect PIN number.

Not supported in the 3G Modem(HC25,HC28)

1.4.20 SimGetLockStatus

Read the Lock status of SIM Card.

```
MODEM_SIM_PIN_AUTH_STATUSSimGetLockStatus (
    refbool Active
);
```

Parameters

Active

parameter which will whether or not save Lock settings of SIM Card.

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

If in the status of Lock, every time applies the power to modem, it should be in the status of SIM_PIN, and must have to enter PIN number.

1.4.21 SimLockUnlock

Set the SIM Card's Lock/Unlock.

```
MODEM_SIM_PIN_AUTH_STATUSSimLockUnlock (
    stringPIN,
    boolbLock
);
```

Parameters

PIN

PIN number of SIM Card.

bLock

Lock setting or not.

True=lock, False=unlock

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

If in the status of Lock, every time applies the power to modem, it should be in the status of SIM_PIN, and must have to enter PIN number.

1.4.22 GetNetworkRegistrationStatus

Read the connection status to network.

MODEM_NETWORK_REGISTRATION_STATUSGetNetworkRegistrationStatus ();

Parameters

None

Return Values

Modem status of current connection to network.

1.4.23 RasDial

Try to RAS connection.

```
BOOLRasDial (
    stringEntryName,
    stringUserName,
    string Password,
    IntPtrhWnd
);
```

Parameters

EntryName

name of phone-book entry, which will be used at phone.
It must to be created before call RasDial function.

UserName

User name that will be used for Remote Access Server dial-up authentication.

Password

password that will be used for Remote Access Server access authentication.

hWnd

user window handle that will be received RAS dial event.

Return Values

True: function performed properly.

False: function performed failed.

Notes

Using RAS dial event to distinguish that wehter accessed or not to RAS, and occurs the same RAS dial event as Microsoft standard API RasDial.

If set the register value of ₩HKEY_CURRENT_USER₩Software₩ATID₩EnableCheckPPAdapter as 1(Default value 0), and if failed to connect to RAS, then will re-try again ten timesat most.

1.4.24 RasHangUp

Remove the RAS Connection.

MODEM_RESULTRasHangUp ();

Parameters

None

Return Values

None

2 C/C++ API Reference

2.1 Enumerations

2.1.1 MODEM_RESULT

The result of a call to functions.

- **MODEM_RESULT_SUCCESS**
  Function executed successfully.
- **MODEM_RESULT_OUTOFMEMORY**
  Failed to assign memory.
- **MODEM_RESULT_INVALID_ARGS**
  Invalid parameter.
- **MODEM_RESULT_UNSUPPORTED**
  Not support command currently.
- **MODEM_RESULT_FAILURE**
  Command execution failed
- **MODEM_RESULT_ALREADY_OPENED**
  Modem port already opened.
- **MODEM_RESULT_NO_BATTERY**
  Main battery insufficiency.
- **MODEM_RESULT_TIMEOUT**
  Module not response, after opened.
- **MODEM_RESULT_ALREADY_ALLOCATED**
  Modem memory already assigned.
- **MODEM_RESULT_NOT_OPENED**
  Call the function without Open.
- **MODEM_RESULT_NOT_POWER_ON**
  Execute function without turning on the power.
- **MODEM_RESULT_ALREADY_POWER_ON**
  Modem power already turned on.

2.1.2 MODEM_SYS_INDEX_USER_WND

Flag that will be transferred by parameter while running Callback function.

- **MODEM_SYS_USER_WND_BASIC**

the basic function such as assign memory, power control and so on.

- **MODEM_SYS_USER_WND_CALL**
  Voice communication function.
- **MODEM_SYS_USER_WND_SMS**
  Text message function.
- **MODEM_SYS_USER_WND_SOCKET**
  Single Socket function.
- **MODEM_SYS_USER_WND_ALL**
  Whole function.

2.1.3 MODEM_SYS_NOTI

reason why Callback delegate is executed.

- **NETWORK_NOT_REGISTERED**
  Not able to check connection to Network for about two minutes.
- **NETWORK_REGISTERED**
  Connection to Network is confirmed.
- **NOCARRIER**
  Disconnected.
- **PORT_CLOSE_FAIL**
  Failed to close Modem port
- **PORT_CLOSE_SUCCESS**
  Succeeded to close Modem port.
- **PORT_OPEN_FAIL**
  Fail to open Modem port.
- **PORT_OPEN_SUCCESS**
  Succeeded to open Modem port.
- **POWER_DOWN_FAIL**
  Failed to turn off Modem power.
- **POWER_DOWN_SUCCESS**
  Succeeded to turn off Modem power.
- **POWER_UP_FAIL**
  Failed to turn on Modem power.
- **POWER_UP_SUCCESS**
  Succeeded to turn on Modem power.
- **POWER_UP_INITIALIZE**
  Start to turn on Modem power.

2.1.4 MODEM_POWER_STATUS

Using ModemGetPowerStatus function to checking the status of the modem power.

- **MODEM_POWER_STATUS_OFF**
  Modem power is in the status of turned OFF.
- **MODEM_POWER_STATUS_ON**
  Modem power is in the status of turned ON.
- **MODEM_POWER_STATUS_INITIALIZING**
  Modem power is in the status of initializing (Not used)
- **MODEM_POWER_STATUS_DEINITIALIZING**
  Modem power is in the status of de-initializing (Not used)
- **MODEM_POWER_STATUS_REINITIALIZING**
  Modem power is in the status of re-initializing (Not used)

2.1.5 MODEM_SYS_PIN_AUTH_STATUS

Responding to SIM Card operation

- **CARD_HOLDER_TRAY_REMOVED**
  SIM Card is not inserted into slot.
- **INCORRECT_PASSWORD**
  Password is incorrect.
- **INVALID_INPUT_VALUE**
  Inputted value is invalid.
- **SIM_BLOCKED**
  Aborted because of a previous operation.
- **SIM_BUSY**
  Previous operation is progressing.
- **SIM_FAILURE**
  Failed to execute SIM relevant function.
- **SIM_INSERTED**
  SIM Card inserted into slot.
- **SIM_PIN**
  the status of waiting to input PIN.
- **SIM_PUK**
  After failed to input PIN in three times, then the status of waiting to input PUK.
- **SIM_READY**
  PIN is already inputted.
- **SIM_SUCCESS**
  Succeeded to perform SIM relevant function.
- **SIM_WRONG**

Invalid SIM Card.

2.1.6 MODEM_NETWORK_REGISTRATION_STATUS

Using ModemGetNetworkRegistrationStatus function to checking the network connection status.

- **COMMAND_FAILED**
  failed to perform command.
- **NOT_REGISTERED**
  in the status of disconnect to network, and do not try to connect to other new network. That reason might as below:
  - no SIM card available
  - no PIN entered
  - no valid Home PLMN entry found on the SIM
- **NOT_REGISTERED_BUT_SEARCHING_OPERATOR**
  searching for aavailable network, although didn't access to network.
- **REGISTERED_TO_GSM**
  connected to GSM(2G) network.
- **REGISTERED_TO_UTRAN**
  connected to UTRAN(3G) network.
- **REGISTRATION_DENIED**
  fail to registration or authentication, The reason might as below:
  - IMSI unknown at HLR
  - illegal Mobile Station
  - illegal Mobile Equipment
- **UNKNOWN**
  unknown.

2.2 Structures

2.2.1 MODEM_CALLBACK_DATA

the structure of indicate the status changing of modem or the results of function execution.

```
typedefstruct
{
  LPARAMlParam;
  intm_UserWnMsg;
  MODEM_SYS_INDEX_USER_WNDnWndIndex;
  WPARAMwParam;
}MODEM_CALLBACK_DATA;
```

- **lParam**
  For more information on message. (Not used)
- **m_UserWnMsg**
  Indicate the call reason of callback.
- **nWndIndex**
  Indicate the call reason of callback. (Not used)
- **wParam**
  For more information on message. (Not used)

2.3 Callback function definition

2.3.1 MODEMCALLBACK

Callback function, that will process data while get reply from modem

Using ModemSetCallback(MODEMCALLBACKpFunc) function to register callback function will be executed which registered when received the structure of indicate the status changing of modem or the results of function execution.

```
typedefvoid (CALLBACK* MODEMCALLBACK)(
    MODEM_SYS_INDEX_USER_WNDnWndIndex,
    intm_UserWnMsg,
    BOOLbBatteryDetect,
    WPARAMwParam,
    LPARAMlParam,
);
```

2.4 Methods

2.4.1 AllocModemContext

Modem devices are required to allocate resources.

MODEM_RESULTAllocModemContext();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if successfully allocate resources.

Notes

It must firstly to be run in order to use Modem API.

2.4.2 DeallocModemContext

Deallocate the modem device resources.

MODEM_RESULTDeallocModemContext();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if successfully deallocate resources.

2.4.3 ModemPowerUp

Turn the modem power on.

MODEM_RESULTModemPowerUp();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

Notes

After Power is applied, it takes a few seconds to initialize Modem. Status is informed to Callback Function after Initialization is complete.

2.4.4 ModemPowerDown

Remove the modem power.

```
MODEM_RESULTModemPowerDown();
```

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

Notes

If removal of Modem power is complete, status is informed to Callback function.

2.4.5 ModemPowerEnable

Enabling or removing power of Modem device synchronously.

```
MODEM_RESULT    ModemPowerEnable(
        BOOL Enable
);
```

Parameters

Enable

Status to be applied to Modem.

TRUE : On, FALSE: Off

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

Notes

Unlike function of Powerup, Powerdown, function will be returned only after status of Modem power completely changed.

2.4.6 ModemSetCallback

Setting Callback function which will be informed the status of modem or the results of the function execution.

```
MODEM_RESULTModemSetCallback(
        MODEMCALLBACKpFunc
);
```

Parameters

pFunc

Callback function which will be performed when received the status of modem or the results of the function execution.

Return Values

MODEM_RESULT_SUCCESS will be returned if function performed successfully.

Notes

In application program, in order to get the status of modem or the results of the function Execution, you must register callback delegate using this function.

2.4.7 ModemSetHwnd

Setting window handle which will be informed the status of modem or the results of the function execution.

```
MODEM_RESULTModemSetHwnd(
    HWNDhWnd
);
```

Parameters

hWnd

application window handle which will be performed when received the status of modem or the results of the function execution.

Return Values

MODEM_RESULT_SUCCESS will be returned if function performed successfully.

Notes

To receive response of Modem using not Callback function but Message, use this function.

2.4.8 ModemGetPowerStatus

Read the status of the modem power.

```
MODEM_POWER_STATUSModemGetPowerStatus();
```

Parameters

None

Return Values

The status values of Current modem power(reference to MODEM_POWER_STATUS)

2.4.9 ModemOpen

Open the port which will communicate with modem, and then checking the status of the Network.

MODEM_RESULTModemOpen();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

2.4.10 ModemClose

Close the port which will communicate with the modem.

MODEM_RESULTModemClose();

Parameters

None

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

2.4.11 ModemIsOpened

Return whether or not the modem communication port is open.

BOOLModemIsOpened()

Parameters

None

Return Values

True: Port is open.

False: Port is close.

2.4.12 ModemIsOpen

Return whether or not the modem communication port is open.

Same function as ModemIsOpened function.

BOOL ModemIsOpened()

Parameters

None

Return Values

True: Port is open.

False: Port is close.

2.4.13 ModemGetRSSI

Reading the value of Modem signal intensity.

```
MODEM_RESULT   GetRSSI(
      ref int RSSI
)
```

Parameters

RSSI

Variable in which the value of Modem signal intensity will be stored.

0 : -113 dBm or less

1 : -111 dBm

2..30 : -109... -53dBm

31 : -51dBm or greater

99 : not known or not detectable

Return Values

MODEM_RESULT_SUCCESS will be returned if performed successfully.

2.4.14 ModemIsRegistered

Return whether or not the modem connected to network.

```
BOOLModemIsRegistered()
```

Parameters

None

Return Values

True: modem connected to Network.

False: modem not connected to Network.

Notes

In order to call Rasdial, there is need to check the status of network connection using this function.

2.4.15 ModemSimChangePin

Change the SIM Card's PIN.

```
MODEM_SIM_PIN_AUTH_STATUSModemSimChangePin (
    LPWSTRszOldPin,
    LPWSTRszNewPin
);
```

Parameters

szOldPin

Current PIN number.

szNewPin

New PIN number

Return Values

SIM_SUCCESS will returned if performed successfully,

2.4.16 ModemSimQueryAuthStatus

Checking the status of Network authentication.

```
MODEM_SIM_PIN_AUTH_STATUSModemSimQueryAuthStatus();
```

Parameters

None

Return Values

Return the status value of the Authentication.

Notes

In order to communicate, there is need to checking whether or not the SIM Card's authentication status is READY. If in the status of SIM_PIN, should enter PIN number, and if in the status of SIM_PUK, then should enter PUK number.

2.4.17 ModemSimQueryCardHolderStatus

Returning the equipped status of SIM Card.

```
MODEM_SIM_PIN_AUTH_STATUS ModemSimQueryCardHolderStatus();
```

Parameters

None

Return Values

Returning the equipped status of SIM Card.

Reference to 2.1.5 MODEM_SIM_PIN_AUTH_STATUS

2.4.18 ModemSimEnterPin

Enter the SIM Card PIN.

```
MODEM_SIM_PIN_AUTH_STATUSModemSimEnterPin(
    LPWSTRszPin
);
```

Parameters

szPin

SIM Card PIN number.

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

In order to communicate, there is need to using SimQueryAuthStatus function to check whether or not the status of SIM card's authentication is READY. If the execution result of SimQueryAuthStatus function is SIM_PIN, then should enter PIN number.

2.4.19 ModemSimEnterPuk

Enter the SIM Card's PUK.

```
MODEM_SIM_PIN_AUTH_STATUSModemSimEnterPuk (
    LPWSTRszPuk,
    LPWSTRszNewPin
);
```

Parameters

szPuk

SIM Card PUK number.

szNewPin

new PIN number

Return Values

SIM_SUCCESS will be returned.

Notes

In order to communicate, there is need to using ModemSimQueryAuthStatusfunction

to check whether or not the status of SIM card's authentication is READY. If continually entered incorrect PIN number three times, then not only should be SIM_PUK status, but also need to enter New PIN number and PUK number.

2.4.20 ModemSimGetPinCounter

Read PIN counter

```
MODEM_SIM_PIN_AUTH_STATUSModemSimGetPinCounter (
    int*pnPinCounter
);
```

Parameters

pnPinCounter

the remaining frequency of enter PIN number.

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

PinCounter will reduce one time, every time enter incorrect PIN number.

Not supported in the 3G Modem(HC25)

2.4.21 ModemSimGetLockStatus

Read the Lock status of SIM Card.

```
MODEM_SIM_PIN_AUTH_STATUSModemSimGetLockStatus (
    BOOL*pbActive
);
```

Parameters

pbActive

parameter which will whether or not save Lock settings.

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

If in the status of Lock, every time applies the power to modem, it should be in the status of SIM_PIN, and must have to enter PIN number.

2.4.22 ModemSimLockUnlock

Set the SIM Card's Lock/Unlock.

```
MODEM_SIM_PIN_AUTH_STATUSModemSimLockUnlock (
    LPWSTRszPIN,
    BOOLbLock
);
```

Parameters

szPIN

PIN number of SIM Card.

bLock

Lock setting or not.

True=lock, False=unlock

Return Values

SIM_SUCCESS will be returned if performed successfully.

Notes

If in the status of Lock, every time applies the power to modem, it should be in the status of SIM_PIN, and must have to enter PIN number.

2.4.23 ModemGetNetworkRegistrationStatus

Read the connection status to network.

```
MODEM_NETWORK_REGISTRATION_STATUSModemGetNetworkRegistrationStatus ();
```

Parameters

None

Return Values

Modem status of current connection to network.

2.4.24 ModemRasDial

Try to RAS connection.

```
BOOLModemRasDial (
    LPRASDIALEXTENSIONSdialExtensions,
    LPTSTRphoneBookPath,
    LPARASDIALPARAMSrasDialParams,
    DWORDNotifierType,
```

LPVOIDnotifier,
LPHRASCONNpRasConn
);

Parameters

dialExtensions

This parameter is ignored and should be set to NULL. On Windows CE, **RasDial** always uses the default behaviors for the **RASDIALEXTENSIONS** options.

phoneBookPath

This parameter should be set to NULL. Dial-up networking stores phone-book entries in the registry rather than in a phone-book file.

rasDialParams

Pointer to a **RASDIALPARAMS** structure that specifies calling parameters for the RAS connection.

NotifierType

Specifies the nature of the *notifier* parameter. If *notifier* is NULL, *NotifierType* is ignored. If *notifier* is not NULL, set *NotifierType* to the following value.

notifier

Pointer to a window handle to receive **RasDial** event notifications. If this parameter is not NULL, **RasDial** sends the window a message for each **RasDial** event. Additionally, the **RasDial** call operates asynchronously: **RasDial** returns immediately, before the connection is established, and uses the window to communicate its progress.

If notifier is NULL, the RasDial call operates synchronously: RasDial does not return until the connection attempt has completed successfully or failed.

pRasConn

Pointer to a variable of type **HRASCONN**. You must set the **HRASCONN** variable to NULL before calling **RasDial**. If **RasDial** succeeds, it stores a handle to the RAS connection into *pRasConn*.

Return Values

Zero indicates success.

Notes

Using RAS dial event to distinguish that wehter accessed or not to RAS, and occurs the same RAS dial event as Microsoft standard API RasDial.

If set the register value of"₩HKEY_CURRENT_USER₩Software₩ATID₩EnableCheckPPPAdapter" as 1(Default value 0), and if failed to connect to RAS, then will re-try again at least ten times.

2.4.25 ModemRasHangUp

RAS Connection release.

```
MODEM_RESULTModemRasHangUp (
    HRASCONN Session
);
```

Parameters

Session

Handle to the remote access connection to terminate. This is a handle returned from a previous call to RasDial or RasEnumConnectionis.

Return Values

Zero indicates success.